Exemplo 4:

Escrever o algoritmo anterior usando o comando PARA em lugar do ENQUANTO.

```
início
|  real: MÉDIA, SOMA;
|  tipo v = vetor [1:80] real;
|  v: NOTA;
|  inteiro: I;
|  SOMA ← 0;
|  leia (NOTA);
|  para I de 1 até 80 passo 1 faça
|  |  SOMA ← SOMA + NOTA [I];
|  fim para;
|  MÉDIA ← SOMA/80;
|  para I de 1 até 80 passo 1 faça
|  |  imprima ("NOTA = ", NOTA [I], "MÉDIA = ", MÉDIA);
|  fim para;
fim.
```

Observe que a leitura das notas foi expressa de uma maneira concisa: *leia* (NOTA). Sempre que o nome de um vetor aparece dissociado de um índice, significa que a operação envolve todos os elementos do vetor.

Exemplo:

leia (NOTA)

equivale a:

```
leia (NOTA [1]);
leia (NOTA [2]);
   :
leia (NOTA [80]);
```

imprima (NOTA);

equivale a:

```
imprima NOTA [1];
imprima NOTA [2];
   :
imprima NOTA [80];
```

Para inicializar todo um vetor com o mesmo valor, também podemos usar o nome de vetor dissociado do índice:

NOTA ← 0 equivale a:

```
NOTA [1] ← 0;
NOTA [2] ← 0;
   :
NOTA [80] ← 0;
```


Para inicializar um vetor com valores diferentes, utilizamos um comando *"construtor"*.

NOTA ← (5.7, 9.5, 10.0, . . . , 7.5);

que equivale a:

```
NOTA [1] ←  5.7;
NOTA [2] ←  9.5;
NOTA [3] ← 10.0;
   :
NOTA [80] ← 7.5;
```

Exemplo 5:

Dado o vetor CRR de caracteres abaixo,

	1	2	3	4	5	6	7	8
CRR	!	U	O	T	R	E	C	A

qual será a sua configuração depois de executados os comandos:

```
para I de 2 até 4 passo 1 faça
|     AUX ← CRR [I];
|     CRR [I] ← CRR [8 – I + 1];
|     CRR [8 – I + 1] ← AUX;
fim para;
AUX ← CRR [1];
CRR [1] ← CRR [8];
CRR [8] ← AUX;
```

Solução:

	1	2	3	4	5	6	7	8
CRR	A	C	E	R	T	O	U	!

Exemplo 6:

Dados dois vetores *R* [1 : 10] e *S* [1 : 20], escreva um algoritmo que:

leia os vetores *R* e *S* a partir de cartões e:

a) gere o vetor *X* correspondente à *união* dos vetores *R* e *S*.
b) gere o vetor *Y* com os elementos *comuns* de *R* e *S*.
c) gere o vetor *Z* com os elementos de *R* que *não estão* em *S*.
d) criar o vetor *W* em que:

```
W [1]  ← R [1]
W [2]  ← S [1]
W [3]  ← R [2]
W [4]  ← S [2]
W [5]  ← R [3]
   :
W [19] ← R [10]
W [20] ← S [10]
W [21] ← S [11]
W [22] ← S [12]
   :
W [30] ← S [20]
```